Lecture d'article

ADAM pour le Deep Learning

Groupe :

ACHIQ Aya
CLETZ Laura
EL MAZZOUJI Wahel

Octobre 2025

Table des matières

1 Introduction

2 Adam et les méthodes adaptatives

2.1 Contexte et motivation

L'apprentissage profond repose sur la minimisation stochastique d'une fonction de coût bruitée. La descente de gradient stochastique et ses variantes à momentum sont simples mais sensibles au choix du taux d'apprentissage et instables en présence de gradients bruités ou clairsemés. Pour pallier ces limites, Adam (*Adaptive Moment Estimation*) a été proposé par Kingma et Ba (2014). L'idée est de combiner les avantages du momentum et de l'adaptation du taux d'apprentissage, comme dans AdaGrad et RMSProp, afin d'obtenir une descente plus rapide et plus stable.

Cependant, plusieurs travaux ont montré que la rapidité de convergence des méthodes adaptatives ne garantit pas toujours une meilleure généralisation. Wilson et al. (2017) soulignent que ces méthodes peuvent conduire à des minima différents de ceux trouvés par SGD, parfois moins performants sur les données de test.

2.2 Principe de l'algorithme Adam

L'algorithme Adam combine deux idées majeures : l'accumulation du momentum et l'adaptation du taux d'apprentissage pour chaque paramètre. À chaque itération t, le gradient stochastique $g_t = \nabla_\theta f_t(\theta_{t-1})$ sert à mettre à jour deux moyennes mobiles : $m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$ et $v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2$, où m_t représente la moyenne et v_t la variance du gradient. Les coefficients usuels sont $\beta_1 = 0.9$ et $\beta_2 = 0.999$. Comme m_0 et v_0 sont initialisés à zéro, Kingma et Ba introduisent une correction de biais : $\hat{m}_t = m_t/(1-\beta_1^t)$ et $\hat{v}_t = v_t/(1-\beta_2^t)$.

La mise à jour s'écrit

$$\theta_t = \theta_{t-1} - \alpha \, \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon},$$

où α est le taux d'apprentissage et $\varepsilon \approx 10^{-8}$ évite la division par zéro. Cette règle combine la direction moyenne du gradient (momentum) et une normalisation par la variance locale, assurant une descente rapide et stable face aux variations d'échelle.

2.3 Interprétation et propriétés

Adam combine momentum et adaptation du pas : m_t lisse la direction de descente, tandis que v_t ajuste le pas selon la variance locale du gradient, réduisant les oscillations et stabilisant l'entraînement.

Adam présente plusieurs propriétés remarquables. Il est invariant à l'échelle des gradients, son pas est borné et contrôlé par α, et il reste peu sensible aux hyperparamètres. Kingma et Ba montrent, dans le cadre convexe, un regret en $\mathcal{O}(\sqrt{T})$, équivalent à AdaGrad mais plus stable sur des objectifs non stationnaires.

Ces propriétés expliquent la popularité d'Adam pour l'apprentissage de réseaux profonds, où les gradients peuvent être bruités, corrélés ou fortement déséquilibrés selon les paramètres.

2.4 Forces et apports de la méthode Adam

Les expériences de Kingma et Ba (2014) sur diverses tâches, telles que la régression logistique et l'apprentissage de réseaux profonds, montrent qu'Adam combine la rapidité d'AdaGrad et

la stabilité de RMSProp. Sur MNIST, il atteint une faible perte en peu d'itérations et une précision comparable à SGD avec momentum, sans réglage manuel du taux d'apprentissage.

L'algorithme se distingue par sa robustesse aux hyperparamètres, sa rapidité de convergence et son efficacité face à des gradients bruités. Les valeurs par défaut ($\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$) offrent de bonnes performances dans la plupart des contextes, ce qui explique sa popularité comme méthode d'optimisation de référence en apprentissage profond.

Enfin, la variante AdaMax, fondée sur la norme infinie, améliore la stabilité numérique tout en conservant la simplicité d'Adam.

2.5 Limites théoriques

Bien qu'efficace en pratique, Adam présente une généralisation limitée. Wilson et al. (2017) montrent que les méthodes adaptatives diffèrent des approches non adaptatives comme SGD.

Théoriquement, Wilson et al. (2017) montrent que SGD converge vers une *solution à norme minimale* ($\|w\|_2$ faible), gage d'une bonne généralisation, tandis que les méthodes adaptatives tendent vers des *solutions à norme infinie minimale* ($\|w\|_\infty$ faible), souvent associées à un surapprentissage.

Wilson et al. (2017) montrent qu'Adam peut converger vers des solutions mal généralisées, alors que SGD atteint la solution optimale, ce qui explique la moindre généralisation des méthodes adaptatives.

Ces travaux soulignent ainsi le compromis fondamental entre rapidité d'apprentissage et capacité de généralisation propre aux méthodes adaptatives.

A Annexes

A.1 Rappels sur les méthodes d'optimisation

A.1.1 Gradient non adaptatif

Une méthode non adaptative utilise un taux d'apprentissage fixe η identique pour tous les paramètres et à toutes les itérations :

$$\theta_{t+1} = \theta_t - \eta\, g_t,$$

où $g_t = \nabla_\theta f_t(\theta_t)$ est le gradient de la fonction de coût.

A.1.2 Gradient adaptatif

Une méthode adaptative ajuste dynamiquement le taux d'apprentissage pour chaque paramètre en fonction de l'historique de ses gradients. Les paramètres dont les gradients varient fortement reçoivent un pas plus petit, ceux dont les gradients sont faibles un pas plus grand :

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t} + \epsilon}\, g_t,$$

avec v_t la moyenne des carrés des gradients passés.

A.1.3 Momentum

Le momentum ajoute une mémoire du gradient passé pour lisser la trajectoire de la descente :

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t.$$

Ce mécanisme stabilise la descente et permet d'atteindre plus facilement le minimum.

Références

[1] D. P. Kingma and J. Ba. Adam : A method for stochastic optimization. *arXiv preprint arXiv :1412.6980*, 2014.

[2] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht. The marginal value of adaptive gradient methods in machine learning. *arXiv preprint arXiv :1705.08292*, 2017.